

AD-A075 576

STANFORD UNIV CA DEPT OF OPERATIONS RESEARCH

F/G 9/2

DOCUMENTATION OF A COMPUTER PROGRAM FOR HILLIER'S HEURISTIC PRO--ETC(U)

JUL 79 N E JACQMI

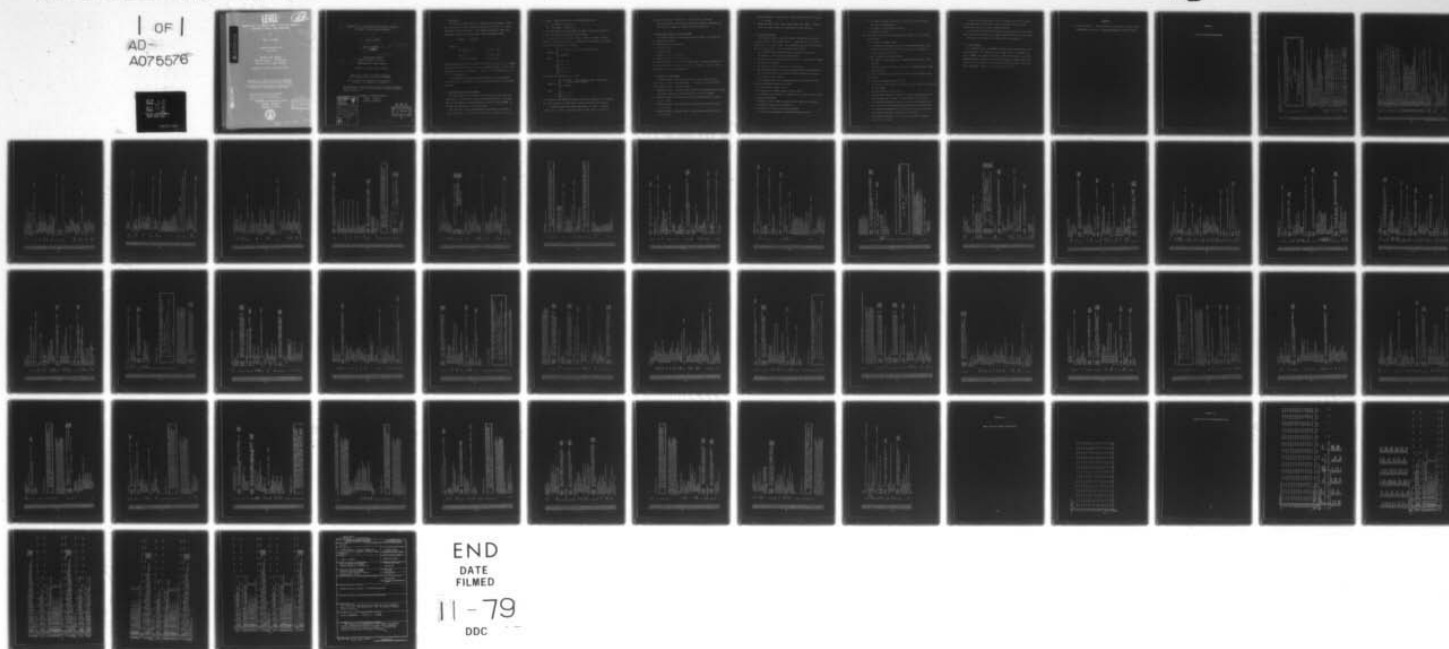
N00014-76-C-0418

UNCLASSIFIED

TR-88

NL

| OF |
AD-A
A075576



END
DATE
FILMED

11-79

DDC

LEVEL II

12

DOCUMENTATION OF A COMPUTER PROGRAM FOR HILLIER'S HEURISTIC
PROCEDURE IN INTEGER LINEAR PROGRAMMING

BY

NANCY E. JACQMIN

TECHNICAL REPORT NO. 88

JULY 1979

PREPARED UNDER CONTRACT

N00014-76-C-0418 (NR-047-061)

FOR THE OFFICE OF NAVAL RESEARCH

Frederick S. Hillier, Project Director

Reproduction in Whole or in Part is Permitted
for any Purpose of the United States Government

This document has been approved for public release
and sale; its distribution is unlimited.

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

DEPARTMENT OF OPERATIONS RESEARCH

STANFORD UNIVERSITY

STANFORD, CALIFORNIA



DDC
RECEIVED
OCT 29 1979
D

79 10 26 019

DDC FILE COPY

AD A075576

6 DOCUMENTATION OF A COMPUTER PROGRAM FOR HILLIER'S HEURISTIC
PROCEDURE IN INTEGER LINEAR PROGRAMMING.

by

10 NANCY E. JACOMIN

9 TECHNICAL REPORT NO. 88

11 JUL 1979

14 TR-98

12 58
15 PREPARED UNDER CONTRACT

N00014-76-C-0418 (NR-047-061)

✓ NSP-MCS 76-81259
FOR THE OFFICE OF NAVAL RESEARCH

Frederick S. Hillier, Project Director

Reproduction in Whole or in Part is Permitted
for any Purpose of the United States Government

This document has been approved for public release
and sale; its distribution is unlimited.

This research was supported in part by National Science Foundation
Grant NSF Number - MCS. 76-81259 A01, Dept. of Operations Research
Stanford University.

DEPARTMENT OF OPERATIONS RESEARCH

STANFORD UNIVERSITY
STANFORD, CALIFORNIA

Accession For	
NTIS GRA&I	
DDC TAB	
Unannounced Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

DDC
RECEIVED
OCT 29 1979
D

402 766

mt

1. Introduction

The computer code HEUR, written in FORTRAN by Bruce H. Faaland to implement the heuristic procedure developed by Frederick S. Hillier [1], is documented and listed in this report. This program is designed to find a good approximate solution to the pure integer linear programming problem,

$$\text{maximize} \quad x_0 = \sum_{j=1}^n c_j x_j,$$

subject to

$$(1) \quad \sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i = 1, 2, \dots, m),$$

$$(2) \quad x_j \geq 0 \quad (j = 1, 2, \dots, n),$$

$$(3) \quad x_j \text{ is an integer} \quad (j = 1, 2, \dots, n).$$

This problem may also be written as $\max cx$ subject to $Ax \leq b$, $x \geq 0$, x integer where the constraint matrix A is $m \times n$, the right hand side b is an $m \times 1$ column vector, the cost coefficients c form a $1 \times n$ row vector and x is an $n \times 1$ column vector.

It is assumed that the set of points satisfying (1) and (2) possesses an interior point. See Hillier [1] for a detailed discussion of this heuristic procedure.

2. Main Program and Input Requirements

The main program for the heuristic code HEUR serves to coordinate the three phases of the search for the desired approximate solution. Any combination of the methods for these phases may be accessed by the programmer via the IFLAG and LFLAG arrays to be described later.

The first card of input contains NAME, any alphanumeric characters to identify the problem. The second card contains the variables M and N in

2I5 format. These variables have the following significance:

M = the number of rows in A.

N = the number of columns in A.

The third input card contains RPT, the number of extra (not necessarily distinct) solutions to be obtained, in F5.0 format.

The fourth input card contains IFLAG(1), IFLAG(2), IFLAG(3), IFLAG(4), LFLAG(1,1), LFLAG(1,2), LFLAG(2,1), and LFLAG(2,2) in 8I4 format. The significance of these variables is as follows:

$$IFLAG(I) = \begin{cases} 1, & \text{if criterion } I' \text{ is to be used in Phase 2} \\ 0, & \text{otherwise} \end{cases}$$

where

$$I' = \begin{cases} B, & \text{if } I = 1 \\ C, & \text{if } I = 2 \\ A, & \text{if } I = 3 \\ D, & \text{if } I = 4. \end{cases}$$

A, B, C and D are as defined in [1, p. 608].

$$LFLAG(I,J) = \begin{cases} 1, & \text{if method } I \text{ is to be used in Phase 1 and method} \\ & J' \text{ is to be used in Phase 2} \\ 0, & \text{otherwise} \end{cases}$$

where

$$J' = \begin{cases} 3, & \text{if } J = 1 \\ 2, & \text{if } J = 2. \end{cases}$$

A description of the methods for Phases 1 and 2 may be found in [1, pp. 605-607].

Note that any number of combinations can be run as specified on this card.

The remaining input cards contain the arrays A, b and c in 15F5.0 format. The A matrix is read in one row at a time, then b is read in

starting on a new card, followed by c also starting on a new card.

Another problem may then be input using the card sequence described above.

The input for a sample run of HEUR is given in Appendix II.

3. Restrictions Relevant to the Use of HEUR

As currently written, the following restrictions apply to the input data and problem size for HEUR:

1. NAME must have ≤ 20 characters.
2. M must be ≤ 120 .
3. N must be ≤ 120 .
4. IFLAG(I) = 0 or 1, I = 1,2,3,4.
5. LFLAG(I,J) = 0 or 1, I = 1,2; J = 1,2.

To change the bounds on M and N you simply have to modify the dimension of A from A(121,362) to A(m' + 1, n' + 2m' + 2) where m' and n' are the new bounds on M and N respectively.

4. Description of Subprograms

Detailed comments on the specific function of each subroutine may be found in the listing of HEUR in Appendix I. Variables of note are defined in these comments as well. In general terms:

1. SUBROUTINE SIMPLX -- Finds an optimal solution to the stated problem ignoring the integer constraints (this is referred to as the "LP problem").
2. SUBROUTINE PHTW01 -- Executes Phase 2 using variable selection criterion
B. [1, p. 608]
3. SUBROUTINE PHTW02 -- Executes Phase 2 using variable selection criterion
C. [1, p. 608]
4. SUBROUTINE PHTW03 -- Executes Phase 2 using variable selection criterion
A. [1, p. 608]

5. SUBROUTINE PHTW04 -- Executes Phase 2 using variable selection criterion D. [1, p. 608]
6. SUBROUTINES PART1, PART2, PART3, PART4, PART5, PART6, PART7 -- Together execute the parts of Phase 3 as described in [1, pp. 612-616].

5. Description of Output

The following output is generated by HEUR. The NAME of the problem is printed followed by the constraint matrix A, right hand side b and cost coefficients c in 15F8.2 format. The normalization factor for the objective function $(1/(\sum_{j=1}^n c_j^2)^{1/2})$ is also recorded in F10.2 format.

On return from SUBROUTINE SIMPLEX the following are printed out:

1. The solution to the LP problem in 15F8.2 format.
2. The row that each original variable is basic in for this solution (0 means the variable is nonbasic).
3. The indices of the basic variables for this solution.
4. The normalized optimal basis inverse (the basis inverse corresponding to the LP solution when the constraints have been normalized as defined in [1, p. 605]) in 6F13.5 format.

Then for each combination of methods specified by the LFLAG and IFLAG arrays the following information is printed:

1. The methods used in Phases 1 and 2.
2. The criterion for variable selection used in Phase 2.
3. The starting value of ALPHA (α as defined in [1, p. 607]) for the current solution being sought.
4. The Phase 1 solution XTWO (the point which with XONE, the LP solution, determines the search line for Phase 2).
5. The results of the Phase 2 search which include:
 - a) The value of ALPHA where the feasible solution was found.

- b) The number of points on the search line that were moved to during the search (excluding XONE).
 - c) The number of trial solutions examined.
 - d) The Phase 2 solution.
6. The results of the first time through Mode 1 of Phase 3 [1, pp. 611-613] which include:
- a) The number of solution changes.
 - b) The normalized improvement (the actual improvement multiplied by the normalization factor for the objective function) in the objective function value in F8.4 format.
7. A record of Phase 3 execution including:
- a) The number of times through each of SUBROUTINES PART4, PART5, PART6 and PART7.
 - b) The number of solution changes during each of SUBROUTINES PART4, PART5, PART6 and PART7.
8. At the end of execution of the heuristic procedure the following results are printed:
- a) The final approximate integer solution.
 - b) The feasibility test slacks (as defined in Part I, Step 3 of [1, p. 612]) in 15F8.2 format.
 - c) The final objective function value.
 - d) The normalized difference in objective function value between the LP solution (XONE) and the Phase 1 solution (XTWO) in F10.4 format.
 - e) The normalized difference in objective function value between the LP solution and the final approximate integer solution in F10.4 format.
 - f) The normalized improvement in the objective function value since the first time through Mode 1 of Phase 3 in F10.4 format.
 - g) The normalized improvement in the objective function value during Phase 3 in F10.4 format.

If $RPT > 0.0$ the information in items 1 through 8 above will be printed out for each extra solution found, as well as the initial one, for every combination of Phase 1 and Phase 2 methods specified.

If more than one problem was included in the input data all of the information above will be printed out for each problem starting with its input NAME.

The output from a sample run of HEUR is given in Appendix III.

6. Further Comments

As currently written, the FORTRAN code HEUR contains approximately 2,300 source statements. All significant array storage is done in a common block labeled COMMON. When compiled on Stanford's IBM 370/168 computer under IBM's FORTRAN-H-Extended compiler, with level two optimization, approximately 512K bytes of core are required for all instruction and data storage. The code is WATFIV compatible. HEUR was last revised in October 1978.

REFERENCE

1. Hillier, Frederick S., "Efficient Heuristic Procedures for Integer Linear Programming with an Interior," Operations Research 17, 600-637 (1969).

APPENDIX I

LISTING OF COMPUTER PROGRAM HEUR


```

120. WRITE(6,503)
121. DO 152 I=1,M
122.   WRITE(6,504)(A(I,J),J=1,N)
123.   WRITE(6,504)
124.   WRITE(6,505)(B(I),I=1,M)
125.   WRITE(6,505)
126.   WRITE(6,506)(C(I),I=1,N)
127.
128. ***** NORMALIZE COEFFICIENTS OF PROBLEM. *****
129.
130. DO 400 I=1,M
131.   SUM=0.0
132.   DO 401 J=1,N
133.     SUM=SUM+A(I,J)*A(I,J)
134.     SUM=SQRT(SUM)
135.   DO 402 J=1,N
136.     A(I,J)=A(I,J)/SUM
137.   CONTINUE
138.   R(I)=R(I)/SUM
139.   CONTINUE
140.   CNORM=0.0
141.   DO 410 J=1,N
142.     CNORM=CNORM+C(J)*C(J)
143.   CONTINUE
144.   CNORM=1.0/SQRT(CNORM)
145.   WRITE(6,510)CNORM
146.
147. ***** THE FOLLOWING SETS UP THE DATA FOR LP ROUTINE. *****
148.
149. LK=N+1
150. LL=M+1
151. LM=M+N
152.
153. ***** STORE COST COEFFICIENTS IN FIRST ROW OF A. *****
154.
155. DO 157 I=1,M
156.   I1=M-I+1
157.   I2=M-I+2
158.   LVL=I+1
159.   OSTAR(L,LVL)=B(I)
160.   DO 159 J=1,N
161.     A(I2,J)=A(I1,J)
162.   CONTINUE
163.   OSTAR(I)=0.0
164.   DO 160 J=1,N
165.     A(I,J)=-C(J)
166.   CONTINUE
167.
168. ***** ADD SLACK VARIABLES. *****
169.
170. DO 151 J=LK,LM
171.   A(I,J)=0.0
172. CONTINUE
173. DO 152 I=2,LL
174.   DO 153 J=LK,LM
175.     A(I,J)=0.0
176.   CONTINUE
177. CONTINUE
178. DO 154 I=2,LL
179.

```

```

180. J=N+1-1
181. A(I,J)=1.0
182. CONTINUE
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
205.
206.
207.
208.
209.
210.
211.
212.
213.
214.
215.
216.
217.
218.
219.
220.
221.
222.
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234.
235.
236.
237.
238.
239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.
301.
302.
303.
304.
305.
306.
307.
308.
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.

```

```

242. C
243. DO 9993 I=1,M
244. IF (IUGM(I).LE.0) GO TO 9993
245. DO 9994 J=1,N
246. A(I,J)=A(I,J)
247. DO 9996 K=2,LL
248. H=1+I
249. DO (H,K)=BP(H,K)
250. A(I)=A(I)
251. CONTINUE
252. DO 9997 I=1,LL
253. XSAVE(I)=X(I)
254. ***** STORE L2 SOLUTION IN XONE. *****
255. C
256. DO 155 J=1,N
257. LKRG(J)
258. IF (LGT=0) GO TO 156
259. XONE(J)=0.0
260. GO TO 155
261. XONE(J)=X(LL)
262. CONTINUE
263. WRITE(6,XRR)
264. WRITE(6,S06)
265. WRITE(6,S001)(XONE(I),I=1,N)
266. LL=9+1
267. DO 578 J=1,N
268. IF (VE(J).LE.0) GO TO 579
269. LKA(J)=KB(J)-1
270. GO TO 578
271. LKB(J)=0
272. CONTINUE
273. WRITE(6,S76)
274. WRITE(6,S71)(LKB(J),J=1,N)
275. WRITE(6,S77)
276. WRITE(6,S71)(JM(I),I=2,LL)
277. JJ=1
278. II=0
279. C
280. C ***** RETRIEVE NORMALIZED OPTIMAL BASIS INVERSE. *****
281. C
282. DO 160 J=1,LL
283. DO 161 I=1,LL
284. H=I+1
285. IF (H.LE.LL) GO TO 162
286. II=1
287. JJ=JJ+1
288. CONTINUE
289. A(I,N+J)=BR(H,I,JJ)
290. CONTINUE
291. CONTINUE
292. DO 163 I=1,M
293. DO 164 J=1,M
294. BR(I,J)=A(I+1,N+J+1)
295. CONTINUE
296. CONTINUE
297. WRITE(6,S202)
298. WRITE(6,S001)((BR(I,J),J=1,M),I=1,M)
299. CONTINUE
300. C

```

```

300.      C      IFLAG(1)=1 MEANS TO USE CRITERION 1 IN PHASE 2.      .....
301.      C      IF AG(I,J)=1 MEANS TO USE METHOD 1 IN PHASE 1 AND .....
302.      C      METHOD J IN PHASE 2.      .....
303.      C
304.      C      COUNT=COUNT+1.0
305.      C      ICOUNT=ICOUNT
306.      C      IF (IFLAG(ICOUNT).EQ.0) GO TO 1932
307.      C      KDH=1
308.      C      KDH1=1
309.      C      IF (LFLAG(KDH,KDH1).EQ.0) GO TO 1999
310.      C      GO TO 169
311.      C      KDH=2
312.      C      KDH1=2
313.      C      IF (LFLAG(KDH,KDH1).EQ.0) GO TO 1999
314.      C      GO TO 169
315.      C      KDH=2
316.      C      KDH1=1
317.      C      IF (LFLAG(KDH,KDH1).EQ.0) GO TO 1999
318.      C      GO TO 169
319.      C      KDH=2
320.      C      KDH1=2
321.      C      IF (LFLAG(KDH,KDH1).EQ.0) GO TO 1999
322.      C      CONTINUE
323.      C      DMK=KDH
324.      C      DM1=KDH1
325.      C      ALPHA=0.0
326.      C      REPEAT=0.0
327.      C      CONTINUE
328.      C
329.      C      ..... PRINT OUT OPTIONS TO BE USED AND PRESENT .....
330.      C      ..... VALUE OF ALPHA. FOR EACH INPUT PROBLEM THIS .....
331.      C      ..... SECTION WILL BE ENTERED (NPT+1) TIMES. ....
332.      C
333.      C      WRITE(6,573)DM,DM1
334.      C      WRITE(6,511)COUNT
335.      C      WRITE(6,509)ALPHA
336.      C      LL=M+1
337.      C      N1=1,LL
338.      C      N1=XSAVE(1)
339.      C
340.      C      .....
341.      C      ..... THIS IS PHASE 1. FIND A POINT NEAR XONE BUT WELL WITHIN THE .....
342.      C      ..... INTERIOR OF THE FEASIBLE LP REGION WHICH CAN BE ROUNDED TO THE .....
343.      C      ..... NEAREST INTEGER SOLUTION WITHOUT VIOLATING THE CONSTRAINTS THAT .....
344.      C      ..... ARE BINDING AT XONE. ....
345.      C      .....
346.      C      .....
347.      C      .....
348.      C      .....
349.      C      IF (DM.EQ.2.0) GO TO 170
350.      C
351.      C      ..... THIS IS METHOD 1 OF PHASE 1. REPLACE H(1) BY .....
352.      C      .....
353.      C      ..... 3*H(1) = 3(1) - 1/2(SUM OVER J IN 1.....N AND .....
354.      C      ..... BASIC AT XONE ABST(1,J)) .....
355.      C      ..... FOR ALL I BINDING AT XONE. ....
356.      C
357.      C      GO 172 I=1,M
358.      C      LL=KB(N+1)
359.      C      IF (LL.LE.0) GO TO 11

```



```

420. Z=REPEAT
421. IF (COUNT.GT.4.0) RETURN
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.

```

```

.....
* THIS IS ENTRY TO PHASE 2. MOVE SLOWLY DOWN THE LINE SEGMENT
* JOINING XONE AND XTWO SEARCHING FOR A NEARBY FEASIBLE INTEGER
* SOLUTION.
.....

IF (COUNT.EQ.1.0) CALL PHTW01
IF (COUNT.EQ.2.0) CALL PHTW02
IF (COUNT.EQ.3.0) CALL PHTW03
IF (COUNT.EQ.4.0) CALL PHTW04

***** PRINT OUT RESULTS OF PHASE 2 SEARCH. *****

WRITE(6.477)
WRITE(6.485) ALPHA
WRITE(6.486) LK
WRITE(6.487) LJ
WRITE(6.502)
WRITE(6.500) (X(LML), LML=1,N)

***** STORE PHASE 2 CJECTIVE VALUE IN ZPH2. *****

ZPH2=0.0
DO 906 LML=1,N
  ZPH2=ZPH2+X(LML)*C(LML)
IF (METHOD.EQ.0) GO TO 1899

.....
* THIS IS PHASE 3. GIVEN THE FEASIBLE SOLUTION FROM PHASE 2. SEARCH
* FOR A BETTER ONE. THIS SEARCH IS CONDUCTED BY TWO ALTERNATING
* MODES.
.....

CALL PART1
IF (NZERO.LF.1) GO TO 903
CALL PART3
CONTINUE
NPR2=0
NPR5=0
NPR4=0
NPR6=0
NPR7=0
LSOLN2=C
LSOLNA=0
LSOLN5=0
LSOLN6=0
LSOLN7=0
CONTINUE
CALL PART2
NPR2=NPR2+1
IF (NPR2.GT.1) GO TO 27

```

```

437. ***** PRINT OUT RESULTS OF FIRST TIME THROUGH *****
438. ***** PHASE 1, PART 2 (CODE 1). *****
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.

```

```

*****
*****
WRITE(C, A72)
WRITE(C, A73)LSULN2
*****
***** STORE INITIAL MODE 1 OBJECTIVE VALUE IN ZPRT2. *****
ZPRT2=0.0
N3 907 LVL=1.N
ZPRT2=ZPRT2+X(LBL)*X(LML)
TEMP=CNORM*(ZPRT2-ZPRT2)
WRITE(C, A75)TEMP
IF(MPRC.LE.1)GO TO 904
CONTINUE
*****
***** STORE CURRENT MODE 1 SOLUTION IN XL. *****
N3 21 J=1.N
XL(J)=X(J)
CONTINUE
J=1
K=N
CONTINUE
CALL PART4
NBR=INBR+1
IF(INVEST.EQ.1)GO TO 43
K=J+1
GO TO 40
*****
***** LOOK TO SEE IF A SIMULTANEOUS CHANGE IN X(JPERM(J))
***** AND X(JPERM(K)) IS FEASIBLE AND INCREASES THE
***** OBJECTIVE FUNCTION VALUE. *****
LL=JPERM(K)
LESS=0
NLESS=0
N3 41 I=1.M
IF(SPRIME(I).GE.0.0) GO TO 41
IF(A(I,LL))43,70,44
LESS=1
GO TO 41
NLESS=2
CONTINUE
LJ=LERS+NLESS
IF(LJ-2)60,50,70
CONTINUE
*****
***** ALL ELEMENTS IN COLUMN JPERM(K) OF A ARE POSITIVE. *****
CALL PART5
NBR5=INBR5+1
IF(IMPROV.EQ.1)GO TO 30
GO TO 70
*****
***** ALL ELEMENTS IN COLUMN JPERM(K) OF A ARE NEGATIVE. *****
CONTINUE
CALL PART6

```

```

540. NBR6=NBR5+1
541. IF (IMPECV.EQ.1) GO TO 30
542.
543. ***** CHECK TO SEE IF THERE ARE MORE K'S TO BE INVESTIGATED FOR *****
544. ***** SIMULTANEOUS CHANGE. *****
545.
546. IF (J.EQ.(K-1)) GO TO 30
547. K=K+1
548. GO TO 40
549. CCNTINUE
550. CALL PART7
551. NBR7=NBR7+1
552.
553. ***** CHECK TO SEE IF THERE ARE MORE J'S TO BE INVESTIGATED. *****
554.
555. NT=N-1
556. IF (NZEPC.LT.NT) NT=NZERO
557. IF (J.EQ.NT) GO TO 100
558. J=J+1
559. K=N TO 20
560. GO TO 20
561.
562. ***** CHECK TO SEE IF MODE 2 HAS FOUND A NEW SOLUTION. *****
563. ***** IF SO RETURN TO MODE 1. *****
564.
565. DO 101 L=1,N
566. IF (X(LML).NE.XL(LML)) GO TO 20
567. CONTINUE
568. CONTINUE
569.
570. ***** PRINT OUT DATA FROM PHASE 3. *****
571.
572. WRITE(6,A68)
573. WRITE(6,A76) NBR2,LSOLN2
574. WRITE(6,A64) NBR4
575. WRITE(6,A54) LSOLN4
576. WRITE(6,A65) NBR5
577. WRITE(6,A55) LSOLN5
578. WRITE(6,A66) NBR6
579. WRITE(6,A56) LSOLN6
580. WRITE(6,A67) NBR7
581. WRITE(6,A57) LSOLN7
582. WRITE(6,A83)
583. WRITE(6,A91)
584. WRITE(6,A500) (X(I),I=1,N)
585. WRITE(6,A508)
586. WRITE(6,A500) (S(I),I=1,M)
587.
588. ***** PRINT OUT FINAL RESULTS. *****
589.
590. Z7=0.0
591. TT=0.0
592. T=0.0
593. DO 102 J=1,N
594. ZZ=Z+C(J)*X(J)
595. TT=C(J)*X(J)+TT
596. T=T+C(J)*XTW(J)
597. CONTINUE
598. WRITE(6,A512) Z7
599. TMD=CND4*(TT-T)
600.

```



```

660.      1340      TOLFIX(I)=INFIX(I)
661.      DO 1308 I=1,3
662.      1308      ZZ(I)=TOL(I)
663.      TCOST=-ABS(TCOST)
664.      PMIX=PRM
665.      M2=4**2
666.      INFS=1
667.
668.      C      ***** CHECK FOR ILLEGAL INPUT. *****
669.      C
670.      C
671.      1371      IF(N) 1304,1304,1371
672.      1372      IF(M-MF)1304,1304,1372
673.      1373      IF(MF-MC)1304,1304,1373
674.      1374      IF(MC)1304,1304,1374
675.      1375      IF(ME-M)1304,1375,1375
676.      C      IF( MOD(INFLA5,4)-1)1400,1320,100
677.      C
678.      C      ***** RECORD INFORMATION ABOUT INITIAL BASIC FEASIBLE SOLUTION.
679.      C      (IT EXISTS DUE TO SET-UP IN MAIN PROGRAM.)
680.      C      ***** JH(I) = J MEANS J IS THE (I-1)ST BASIC VARIABLE
681.      C      INCLUDING SLACKS AND ARTIFICIALS.
682.      C      KB(J) = 1 MEANS VARIABLE J IS BASIC IN ROW (I-1).
683.      C      KB(J) = 0 MEANS J IS NONBASIC.
684.      C      (RECALL THAT THE COST COEFFICIENTS HAVE BEEN INSTALLED IN
685.      C      ROW 1 OF A.)
686.      C
687.      1400      DO 1401 I=1,M
688.      1401      JH(I)=0
689.      C
690.      C      ***** INITIALIZE KB AND JH TO REPRESENT THE INITIAL BASIC
691.      C      ***** FEASIBLE SOLUTION. THIS SOLUTION EXISTS DUE TO
692.      C      ***** SET-UP IN MAIN PROGRAM.
693.      C
694.      C      DO 1402 J=1,N
695.      C      KB(J)=0
696.      C      KQ=0
697.      C      DO 1403 L=MF,M
698.      C      IF(A(L,J))1404,1403,1404
699.      C      KQ=LO+1
700.      C      LO=L
701.      C      CONTINUE
702.      C
703.      C      ***** CHECK WHETHER VARIABLE J IS A CANDIDATE TO BE BASIC. *****
704.      C
705.      C      IF(KQ-1)1402,1405,1402
706.      C      IA=LO
707.      C      IF(JH(IA))1402,1405,1402
708.      C      IF(A(IA,J)*B(IA))1402,1407,1407
709.      C
710.      C      ***** J IS A BASIC VARIABLE IN PC# (IA-1). *****
711.      C      ***** SET JH AND K3 ACCORDINGLY. *****
712.      C
713.      C
714.      C      JH(IA)=J
715.      C      K3(J)=IA
716.      C      CONTINUE
717.      C      CONTINUE
718.      C
719.      1100      ASSIGN 1102 TO K3IV

```


23

```

900.      IR=IT
901.      CONTINUE
902.      IF(NEG)1016,1099,1016
903.
904.      ***** FIND PIVOT AMONG NEGATIVE EQUATIONS. IN WHICH *****
905.      X/Y IS LESS THAN AA, THAT HAS THE LARGEST *****
906.      ABS(Y). STORE THE ARG MAX IN IR. *****
907.
908.      B3=-TPIV
909.      DO 1030 I=MF,M
910.      IF(X(I))1012,1030,1030
911.      IF(Y(I)-BB)1022,1030,1030
912.      IF(Y(I)*AA-X(I)) 1024,1024,1030
913.      BB=Y(I)
914.      IR=I
915.      CONTINUE
916.      IF(IR)207,207,210
917.
918.      ***** NO PIVOT ROW HAS BEEN FOUND FOR COLUMN JT. *****
919.
920.      K=5
921.      IF(PMIX)201,400,201
922.
923.      ***** CHECK IF ITERATION LIMIT IS EXCEEDED. *****
924.
925.      IF(ITER-NCUT) 900,160,160
926.      NUMDV=NUMDV+1
927.      YI=-Y(IR)
928.      V(IR)=-1.
929.
930.      ***** TRANSFORM INVERSE TJ CORRESPOND TO *****
931.      JT BEING BASIC IN ROW (IR-1). *****
932.
933.      DO 904 L=1,M
934.      IF(E(IR,L))905,514,905
935.      CONTINUE
936.      GO TO 904
937.      XY=E(IR,L)/YI
938.      E(IR,L)=0
939.      DO 906 I=1,M
940.      E(I,L)=E(I,L)+XY*Y(I)
941.      CONTINUE
942.
943.      ***** TRANSFORM X TO CORRESPOND TO *****
944.      JT BEING BASIC IN ROW (IR-1). *****
945.
946.      XY=X(IR)/YI
947.      X(IR)=0.
948.      DO 909 I=1,M
949.      X(I)=X(I)+XY*Y(I)
950.      Y(IR)=-YI
951.      GO TO KPIV,(221,1102)
952.
953.      ***** ADJUST JH AND KB TO MAKE JT BASIC IN ROW (IR-1). *****
954.
955.      JA=JH(IF)
956.      IF(IA)213,213,214
957.      KB(IA)=0
958.      KJ(JT)=ID
959.

```

```

360. JH(IP)=JT
361. LAM
362. ITR=ITER+1
363. IVC=IVC+1
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.

```

JH(IP)=JT
 LAM
 ITR=ITER+1
 IVC=IVC+1
 ***** CHECK INVERSION FREQUENCY. *****
 IF(INVC-NVER)1200,1320,1200
 ***** TOO MANY ITERATIONS HAVE BEEN PERFORMED. *****
 K=6
 ***** STORE AX - 7 AT Y. *****
 ASSIGN 410 TO NOEL
 DO 401 I=1,M
 Y(I)=-3(I)
 DO 402 I=1,M
 JA=JH(I)
 IF(JA) 403,402,403
 DO 405 I=1,M
 IF(A(I),JA) 415,405,415
 Y(I)=Y(I)+X(I)*A(I),JA
 CONTINUE
 CONTINUE
 ***** FIND SUM AND MAXIMUM OF ERRORS IN AX - B. STORE *****
 ***** IN TERR(1) AND TERR(2) RESPECTIVELY. *****
 DO 481 I=1,M
 YI=Y(I)
 IF(JH(I)) 472,471,472
 YI=YI+X(I)
 TERR(LA+1)=TERR(LA+1)+ABS(YI)
 IF(ABS(TERR(LA+2))-ABS(YI))482,481,481
 TERR(LA+2)=YI
 CONTINUE
 ***** STORE B TIMES BASIS AT DT. *****
 I=1
 JH=JH(I)
 IF(JH)360,411,300
 ***** FIND SUM AND MAXIMUM OF ERRORS IN PA - 0. STORE *****
 ***** IN TERR(3) AND TERR(4) RESPECTIVELY. *****
 TERR(LA+3)=TERR(LA+3)+ABS(DT)
 IF(ABS(TERR(LA+4))-ABS(DT)) 413,411,411
 TERR(LA+4)=DT
 I=I+1
 IF(I,DT,MIGD TO 5
 GO TO 4
 IF(LA) 163,191,193
 LA=4
 IF(INCLAG-411720,193,193
 IF(K-5)132,194,1392
 ASSIGN 1392 TO KJNY
 GO TO 400


```

1030. XL(J)=XONE(J)+ALPHA*(XTWO(J)-XONE(J))
1081. ZZ=.05
1082. GO TO 10
1083. ALPHA=.0
1084. ZZ=.05
1085. TEMP=.0-ALPHA
1086.
1087. ***** IDENTIFY JOINT ON LINE SEGMENT JOINING XONE AND *****
1088. XTWO THAT HAS BEEN REACHED. STORE IT IN XL. *****
1089. ***** NOTICE THAT AS ALPHA INCREASES WE APPROACH XTWO *****
1090. ***** GOING ALONG SEGMENT JOINING XONE AND XTWO. *****
1091.
1092. DO 15 J=1,N
1093. XL(J)=XONE(J)+ALPHA*(XTWO(J)-XONE(J))
1094.
1095. ***** NOW OBTAIN NEAREST INTEGER POINT. VARIABLES *****
1096. ***** THAT GO NEGATIVE ARE SET TO ZERO. *****
1097.
1098. DO 20 J=1,N
1099. IF (XL(J)<.21).GOTO 21
1100. X(J)=.0
1101. GO TO 20
1102. X(J)=AINT(XL(J)+.5)
1103. CONTINUE
1104.
1105. ***** FIND QMAX. THE MEASURE OF INFEASIBILITY OF X. *****
1106.
1107. DO 10 I=1,M
1108. T=.0
1109. DO 10 J=1,N
1110. T=T+A(I,J)*X(J)
1111. Q(I)=T-B(I)
1112. QMAX=Q(I)
1113. DO 20 I=2,M
1114. IF (QMAX-GT.Q(I))GO TO 20
1115. QMAX=Q(I)
1116. CONTINUE
1117. IF (QMAX-GT.Q(0))GO TO 6
1118. RETURN
1119.
1120. ***** FIND DIRECTION OF FAVORABLE CHANGE FOR EACH *****
1121. ***** X(J) IF ONE EXISTS. LESS AND NLESS INDICATE *****
1122. ***** NEGATIVE AND POSITIVE A(I,J) RESPECTIVELY. *****
1123.
1124. DO 27 J=1,N
1125. NLESS=.0
1126. DO 27 I=1,M
1127. IF (QMAX-GT.Q(I))GO TO 23
1128. IF (A(I,J)>.26).GOTO 26
1129. IF (NLESS.EQ.1)GO TO 26
1130. LESS=.1
1131. GO TO 23
1132. IF (LESS.EQ.1)GO TO 26
1133. NLESS=.1
1134. GO TO 23
1135. DELX(J)=.0
1136. GO TO 22
1137. CONTINUE
1138. IF (LESS.EQ.1)GO TO 30
1139.

```

1140.	IF(X(J).LE.0.0)GO TO 17		
1141.	DELX(J)=-1.0		
1142.	GO TO 22		
1143.	DELX(J)=0.0		
1144.	GO TO 22		
1145.	DELX(J)=1.0		
1146.	CONTINUE		
1147.			
1148.	***** IDENTIFY WHICH VARIABLE WHEN CHANGED WOULD GIVE	*****	
1149.	***** THE LARGEST IMPROVEMENT., STORE INDEX IN K.	*****	
1150.			
1151.	DO 42 J=1,N		
1152.	IF(DELX(J))32,40,35		
1153.	TT=Q(1)-A(1,J)		
1154.	DO 36 I=2,M		
1155.	TT=Q(I)-A(I,J)		
1156.	IF(T.GT.TT)TT=T		
1157.	CONTINUE		
1158.	QSTAR(J)=TT		
1159.	GO TO 42		
1160.	TT=Q(1)+A(1,J)		
1161.	DO 21 I=2,M		
1162.	TT=Q(I)+A(I,J)		
1163.	IF(T.GT.TT)TT=T		
1164.	CONTINUE		
1165.	QSTAR(J)=TT		
1166.	GO TO 42		
1167.	QSTAR(J)=QMAX		
1168.	CONTINUE		
1169.	K=0		
1170.	TT=-1000.34		
1171.	DO 45 J=1,N		
1172.	IF(QMAX.LE.QSTAR(J))GO TO 45		
1173.	TEMP=QMAX-QSTAR(J)		
1174.	IF(TEMP.LE.T)GO TO 45		
1175.	K=J		
1176.	T=TEMP		
1177.	CONTINUE		
1178.	IF(K.EQ.0)GO TO 10		
1179.			
1180.	***** CHANGE X(K) AND RECOMPUTE INFEASIBILITIES.	*****	
1181.	X(K)=X(K)+DELX(K)		
1182.	LJ=LJ+1		
1183.	QMAX=QSTAR(K)		
1184.	IF(DELX(K))54,5.46		
1185.	DO 55 I=1,M		
1186.	Q(I)=Q(I)-A(I,K)		
1187.	CONTINUE		
1188.	GO TO 5		
1189.	DO 48 I=1,M		
1190.	Q(I)=Q(I)+A(I,K)		
1191.	CONTINUE		
1192.	GO TO 5		
1193.			
1194.	***** NO FURTHER REDUCTION IN INFEASIBILITY CAN BE ATTAINED.	*****	
1195.	***** MOVE TO A NEW POINT ON SEARCH SEGMENT.	*****	
1196.			
1197.			
1198.	IF(DH1.EQ.2.0)GO TO 34.0		
1199.			


```

1260. X5(121),SQRTIME(121),X(121),XL(121),XONE(121),XTWO(121)
1261. COMMON A,ALPHA,B,C,CAPR,CPRM,DELX,I,INPROV,DELTA,INVEST,J,
1262. XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,ML,LO,LTMP,M,METHOD,
1263. XNLESS,NZERO,PRI,G,QMAX,QSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
1264. XTWO,TT,X,XONE,XTWO,JSUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
1265. XLSOLN6,LSOLN7
1266. IF(24.EQ.0.0)GO TO 1
1267.
1268. ***** IDENTIFY POINT ON LINE SEGMENT JOINING XONE AND *****
1269. ***** XTWO THAT HAS BEEN REACHED, STORE IT IN XL. *****
1270. ***** NOTICE THAT AS ALPHA INCREASES WE APPROACH XTWO *****
1271. ***** GOING ALONG SEGMENT JOINING XONE AND XTWO. *****
1272.
1273. DO 1900 J=1,N
1274. XL(J)=X(J)+ALPHA*(XTWO(J)-XONE(J))
1275. ZZ=.05
1276. GO TO 10
1277. ALPHA=0.0
1278. ZZ=.05
1279. TEMP=1.0-ALPHA
1280.
1281. ***** IDENTIFY POINT ON LINE SEGMENT JOINING XONE AND *****
1282. ***** XTWO THAT HAS BEEN REACHED, STORE IT IN XL. *****
1283. ***** NOTICE THAT AS ALPHA INCREASES WE APPROACH XTWO *****
1284. ***** GOING ALONG SEGMENT JOINING XONE AND XTWO. *****
1285.
1286. DO 15 J=1,N
1287. XL(J)=XONE(J)+ALPHA*(XTWO(J)-XONE(J))
1288.
1289. ***** NOW OBTAIN NEAREST INTEGER POINT: VARIABLES *****
1290. ***** THAT GO NEGATIVE ARE SET TO ZERO. *****
1291.
1292. DO 820 J=1,N
1293. IF(XL(J)>821.821,821,822
1294. X(J)=0.0
1295. GO TO 820
1296. X(J)=AINT(XL(J)+.5)
1297. CONTINUE
1298.
1299. ***** FIND QMAX, THE MEASURE OF INFEASIBILITY OF X. *****
1300.
1301. DO 19 I=1,M
1302. T=0.0
1303. DO 18 J=1,N
1304. T=T+A(I,J)*X(J)
1305. Q(I)=T-B(I)
1306. QMAX=0.0
1307. DO 530 I=1,M
1308. IF(Q(I).GT.0.0)QMAX=QMAX+Q(I)
1309. CONTINUE
1310. IF(QMAX.GT.0.0)GO TO 6
1311. RETURN
1312. CONTINUE
1313.
1314. ***** IDENTIFY WHICH VARIABLE WHEN CHANGED WOULD GIVE *****
1315. ***** THE LARGEST IMPROVEMENT, STORE INDEX IN K. *****
1316. ***** ALSO IDENTIFY DIRECTION OF FAVORABLE CHANGE FOR *****
1317. ***** EACH X(I). *****
1318.
1319. LL=M

```

```

1320. KEY=0
1321. O1 535 J=1.04
1322. TT=0.0
1323. DO 536 I=1,LL
1324. IF (O(I)) 536, 536, 537
1325. T=T+A(I,J)
1326. CONTINUE
1327. IF (T) 533, 533, 540
1328. DELX(J)=1.0
1329. GO TO 541
1330. IF (X(J).LE.0.0) GO TO 539
1331. DELX(J)=-1.0
1332. GO TO 542
1333. DELX(J)=0.0
1334. LO(J)=0
1335. GO TO 535
1336. DO 543 I=1,LL
1337. T=O(I)+A(I,J)
1338. IF (T.GT.0.0) TT=TT+T
1339. CONTINUE
1340. GO TO 545
1341. DO 544 I=1,LL
1342. T=O(I)-A(I,J)
1343. IF (T.GT.0.0) TT=TT+T
1344. CONTINUE
1345. QSTAR(J)=TT
1346. IF (QSTAR(J).GT.0.0) GO TO 1112
1347.
1348. C
1349. C
1350. C
1351. C
1352. C
1353. C
1354. C
1355. C
1356. C
1357. C
1358. C
1359. C
1360. C
1361. C
1362. C
1363. C
1364. C
1365. C
1366. C
1367. C
1368. C
1369. C
1370. C
1371. C
1372. C
1373. C
1374. C
1375. C
1376. C
1377. C
1378. C
1379. C

537 KEY=0
538 O1 535 J=1.04
539 TT=0.0
540 DO 536 I=1,LL
541 IF (O(I)) 536, 536, 537
542 T=T+A(I,J)
543 CONTINUE
544 IF (T) 533, 533, 540
545 DELX(J)=1.0
546 GO TO 541
547 IF (X(J).LE.0.0) GO TO 539
548 DELX(J)=-1.0
549 GO TO 542
550 DELX(J)=0.0
551 LO(J)=0
552 GO TO 535
553 DO 543 I=1,LL
554 T=O(I)+A(I,J)
555 IF (T.GT.0.0) TT=TT+T
556 CONTINUE
557 GO TO 545
558 DO 544 I=1,LL
559 T=O(I)-A(I,J)
560 IF (T.GT.0.0) TT=TT+T
561 CONTINUE
562 QSTAR(J)=TT
563 IF (QSTAR(J).GT.0.0) GO TO 1112
564
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C

1112 X(J)=X(J)+DELX(J)
1113 RETURN
1114 CONTINUE
1115 IF (TT-QMAX) 546, 547, 547
1116 LO(J)=1
1117 K=J
1118 KEY=KEY+1
1119 GO TO 535
1120 LO(J)=0
1121 CONTINUE
1122 IF (KEY-1) 110, 0.0, 0.07
1123 TEMPE=100034
1124 DO 55 J=1,N
1125 IF (LO(J).EQ.0) GO TO 55
1126 IF (LO(J).NE.0) DELX(J)/(CMAX-QSTAR(J)).LE.TEMPE GO TO 55
1127 JHIGH=J
1128 TEMPE=(C(J)+DELX(J))/(CMAX-QSTAR(J))
1129 CONTINUE
1130 K=JHIGH
1131
1132 C
1133 C
1134 C
1135 C
1136 C
1137 C
1138 C
1139 C
1140 C
1141 C
1142 C
1143 C
1144 C
1145 C
1146 C
1147 C
1148 C
1149 C
1150 C
1151 C
1152 C
1153 C
1154 C
1155 C
1156 C
1157 C
1158 C
1159 C
1160 C
1161 C
1162 C
1163 C
1164 C
1165 C
1166 C
1167 C
1168 C
1169 C
1170 C
1171 C
1172 C
1173 C
1174 C
1175 C
1176 C
1177 C
1178 C
1179 C

9 X(K)=X(K)+DELX(K)
10 L=L+1
11 T=DELX(K)
12 DO 50 I=1,LL
13 O(I)=O(I)+A(I,K)*T
14 QMAX=QSTAR(K)
15 GO TO 5

```



```

1500. ***** IDENTIFY WHICH VARIABLE WHEN CHANGED WOULD *****
1501. GIVE THE LARGEST IMPROVEMENT. STORE *****
1502. INDEX IN KK IF FEASIBILITY IS ATTAINED, IN *****
1503. K IF NOT. ALSO IDENTIFY DIRECTION OF FAVOR *****
1504. ***** ATTLE CHANGE FOR EACH X(J). *****
1505.
1506. LL=M
1507. KEY=0
1508. TL=-10**34
1509. KK=0
1510. DO 535 J=1,N
1511.   TT=0.0
1512.   DO 536 I=1,LL
1513.     IF(Q(I)) 536,536,537
1514.     TT=TT+A(I,J)
1515.     CONTINUE
1516.     TL=ABS(TT)
1517.     IF(T) 538,539,540
1518.     DELX(J)=1.0
1519.     GO TO 541
1520.     IF(X(J)-LE,0.0) GO TO 539
1521.     DELX(J)=-1.0
1522.     GO TO 542
1523.     DELX(J)=0.0
1524.     LO(J)=0
1525.     GO TO 535
1526.   DO 543 I=1,LL
1527.     TT=Q(I)+A(I,J)
1528.     IF(T.GT.0.0) TT=TT+T
1529.     CONTINUE
1530.     GO TO 545
1531.   DO 544 I=1,LL
1532.     TT=Q(I)-A(I,J)
1533.     IF(T.GT.0.0) TT=TT+T
1534.     CONTINUE
1535.     QSTAP(J)=TT
1536.     IF(OSTAP(J).GT.0.0) GO TO 1112
1537.     IF(TL1-LE,TL) GC TO 535
1538.     TL=TL1
1539.     KK=J
1540.     GO TO 535
1541.   CONTINUE
1542.   IF(TT-QMAX) 546,547,547
1543.   LO(J)=1
1544.   K=J
1545.   KEY=KEY+1
1546.   GO TO 535
1547.   LO(J)=0
1548.   CONTINUE
1549.   IF(KK.EQ.0) GO TO 2
1550. ***** FEASIBILITY HAS BEEN ATTAINED. *****
1551. X(KK)=X(KK)+DELX(KK)
1552. RETURN
1553. IF(KEY-1) 10,9,47
1554. TEMPE=-10**34
1555. DO 55 J=1,N
1556.   IF(LO(J).EQ.0) GO TO 55
1557.
1558.
1559.

```

```

1550. IF((QMAX-QSTAR(J)).LE.TEMP)GO TO 55
1561. JHIGH=J
1562. TEMP=QMAX-QSTAR(J)
1563. CONTINUE
1564.
1565.
1566.
1567.
1568.
1569.
1570.
1571.
1572.
1573.
1574.
1575.
1576.
1577.
1578.
1579.
1580.
1581.
1582.
1583.
1584.
1585.
1586.
1587.
1588.
1589.
1590.
1591.
1592.
1593.
1594.
1595.
1596.
1597.
1598.
1599.
1600.
1601.
1602.
1603.
1604.
1605.
1606.
1607.
1608.
1609.
1610.
1611.
1612.
1613.
1614.
1615.
1616.
1617.
1618.
1619.

55      ***** CHANGE X(K) AND RECOMPUTE INFEASIBILITIES. *****
C
C
C      K=JHIGH
X(K)=X(K)+DELX(K)
LJ=LJ+1
Y=DELX(K)
DO 50 I=1,LL
  Q(I)=Q(I)+A(I,K)*Y
QMAX=QSTAR(K)
GO TO 5
CONTINUE
C
C      ***** NO FURTHER REDUCTION IN INFEASIBILITY CAN BE *****
C      ***** ATTAINED. MOVE TO A NEW POINT ON SEARCH *****
C      ***** SEGMENT. *****
C
C      IF(2*H1.EQ.2.0)GO TO 840
C
C      ***** THIS IS PHASE 2. METHOD 1 (METHOD 3 IN REFERENCE *****
C      ***** 1 (P.610)). INCREASE ALPHA BY THE MINIMUM *****
C      ***** AMOUNT REQUIRED TO OBTAIN A DIFFERENT ROUNDED *****
C      ***** SOLUTION THAN AT XL. *****
C
C      DO 830 J=1,N
  TEMP=XTRN(J)-XCNF(J)
  IF(TEMP)831,832,833
  QSTAR(J)=10.0**34
  GO TO 830
  IF(XL(J).LT.0.5)GO TO 832
  QSTAR(J)=-(XL(J)+.5001-AINT(XL(J)+.5))/TEMP
  GO TO 830
  QSTAR(J)=(AINT(XL(J)+.5)+.5001-XL(J))/TEMP
  CONTINUE
  QMAX=10.0**34
  DO 835 J=1,N
    IF(QSTAR(J).LT.QMAX)QMAX=QSTAR(J)
    ALPHA=ALPHA+QMAX
    GO TO 841
C
C      ***** THIS IS PHASE 2. METHOD 2. INCREASE ALPHA BY *****
C      ***** A FIXED INCREMENT ZZ. *****
C
C      ALPHA=ALPHA+ZZ
CONTINUE
LK=LK+1
IF(ALPHA.LE.1.0)GO TO 2
C
C      ***** ALPHA .GT. 1. PHASE 2 HAS FAILED TO FIND A *****
C      ***** FEASIBLE INTEGER SOLUTION. *****
C
C      METHOD=0
C      RETURN
C      END

```

```

1620. SUBROUTINE PHMCOA
1621.
1622.
1623.
1624.
1625.
1626.
1627.
1628.
1629.
1630.
1631.
1632.
1633.
1634.
1635.
1636.
1637.
1638.
1639.
1640.
1641.
1642.
1643.
1644.
1645.
1646.
1647.
1648.
1649.
1650.
1651.
1652.
1653.
1654.
1655.
1656.
1657.
1658.
1659.
1660.
1661.
1662.
1663.
1664.
1665.
1666.
1667.
1668.
1669.
1670.
1671.
1672.
1673.
1674.
1675.
1676.
1677.
1678.
1679.

```

```

.....
THIS IS PHASE 2. CRITERION 3 (CORRESPONDING TO CRITERION 0 IN
REFERENCE 1 (P. 608)). THE 'INFEASIBILITY' OF A SOLUTION X IS
.....
      QMAX = MAX (COMPONENTS OF AX - 9) .
.....
      Q(1) 'MEASURES' IN INFEASIBILITY IN CONSTRAINT 1.
.....
      THE 'IMPROVEMENT' OBTAINED BY CHANGING THE VALUE OF A VARIABLE
      X(J) IS
.....
      (C(J)*(CHANGE IN X(J)))/ - (CHANGE IN QMAX).
.....
      DIMENSION A(121,262),B(121),C(121),CAPR(121,121),CDEF4(121),
      XDELX(121),DELTA(121),JPERM(121),LQ(121),Q(121),OSTAR(121),
      XS(121),SPRIME(121),X(121),XL(121),XONE(121),XTWO(121),
      COMMON A,ALPHA,B,C,CAPR,CDEFM,DELX,I,IMPROV,DELTA,INVEST,J,
      XHIGH,JJ,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTERP,M,METHOD,
      XNLESS,NZERO,PHI,Q,QMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTWO,TT,X,XL,XONE,XTWO,USUBK,ZM,ZZ,N,L,SOLN2,L,SOLN4,L,SOLN5,
      XLSOLN6,L,SOLN7
      IF(ZM.EQ.0.0)GO TO 1
.....
      IDENTIFY POINT ON LINE SEGMENT JOINING XONE AND
      XTWO THAT HAS BEEN REACHED. STORE IT IN XL.
.....
DO 1000 J=1,N
  XL(J)=XONE(J)+ALPHA*(XTWO(J)-XONE(J))
  ZZ=.05
  DO 10 TO 10
    ALPHA=.0
    ZF=.05
    TEMP=1.0-ALPHA
.....
      IDENTIFY POINT ON LINE SEGMENT JOINING XONE AND
      XTWO THAT HAS BEEN REACHED. STORE IT IN XL.
.....
DO 15 J=1,N
  XL(J)=XONE(J)+ALPHA*(XTWO(J)-XONE(J))
.....
      NOW OBTAIN NEAREST INTEGER POINT X. VARIABLES
      THAT GO NEGATIVE ARE SET TO ZERO.
.....
DO 820 J=1,N
  IF(XL(J))21,821,322
  X(J)=0.0
  DO 10 TO 920
    X(J)=AINT(XL(J))+.5
  CONTINUE
.....
      FIND QMAX. THE MEASURE OF INFEASIBILITY OF X.
.....

```

```

1513.
1514.
1515.
1516.
1517.
1518.
1519.
1520.
1521.
1522.
1523.
1524.
1525.
1526.
1527.
1528.
1529.
1530.
1531.
1532.
1533.
1534.
1535.
1536.
1537.
1538.
1539.
1540.
1541.
1542.
1543.
1544.
1545.
1546.
1547.
1548.
1549.
1550.
1551.
1552.
1553.
1554.
1555.
1556.
1557.
1558.
1559.
1560.
1561.
1562.
1563.
1564.
1565.
1566.
1567.
1568.
1569.
1570.
1571.
1572.
1573.
1574.
1575.
1576.
1577.
1578.
1579.
1580.
1581.
1582.
1583.
1584.
1585.
1586.
1587.
1588.
1589.
1590.
1591.
1592.
1593.
1594.
1595.
1596.
1597.
1598.
1599.
1600.
1601.
1602.
1603.
1604.
1605.
1606.
1607.
1608.
1609.
1610.
1611.
1612.
1613.
1614.
1615.
1616.
1617.
1618.
1619.
1620.
1621.
1622.
1623.
1624.
1625.
1626.
1627.
1628.
1629.
1630.
1631.
1632.
1633.
1634.
1635.
1636.
1637.
1638.
1639.
1640.
1641.
1642.
1643.
1644.
1645.
1646.
1647.
1648.
1649.
1650.
1651.
1652.
1653.
1654.
1655.
1656.
1657.
1658.
1659.
1660.
1661.
1662.
1663.
1664.
1665.
1666.
1667.
1668.
1669.
1670.
1671.
1672.
1673.
1674.
1675.
1676.
1677.
1678.
1679.
1680.
1681.
1682.
1683.
1684.
1685.
1686.
1687.
1688.
1689.
1690.
1691.
1692.
1693.
1694.
1695.
1696.
1697.
1698.
1699.
1700.
1701.
1702.
1703.
1704.
1705.
1706.
1707.
1708.
1709.
1710.
1711.
1712.
1713.
1714.
1715.
1716.
1717.
1718.
1719.
1720.
1721.
1722.
1723.
1724.
1725.
1726.
1727.
1728.
1729.
1730.
1731.
1732.
1733.
1734.
1735.
1736.
1737.
1738.
1739.
1740.
1741.
1742.
1743.
1744.
1745.
1746.
1747.
1748.
1749.
1750.
1751.
1752.
1753.
1754.
1755.
1756.
1757.
1758.
1759.
1760.
1761.
1762.
1763.
1764.
1765.
1766.
1767.
1768.
1769.
1770.
1771.
1772.
1773.
1774.
1775.
1776.
1777.
1778.
1779.
1780.
1781.
1782.
1783.
1784.
1785.
1786.
1787.
1788.
1789.
1790.
1791.
1792.
1793.
1794.
1795.
1796.
1797.
1798.
1799.
1800.
1801.
1802.
1803.
1804.
1805.
1806.
1807.
1808.
1809.
1810.
1811.
1812.
1813.
1814.
1815.
1816.
1817.
1818.
1819.
1820.
1821.
1822.
1823.
1824.
1825.
1826.
1827.
1828.
1829.
1830.
1831.
1832.
1833.
1834.
1835.
1836.
1837.
1838.
1839.
1840.
1841.
1842.
1843.
1844.
1845.
1846.
1847.
1848.
1849.
1850.
1851.
1852.
1853.
1854.
1855.
1856.
1857.
1858.
1859.
1860.
1861.
1862.
1863.
1864.
1865.
1866.
1867.
1868.
1869.
1870.
1871.
1872.
1873.
1874.
1875.
1876.
1877.
1878.
1879.
1880.
1881.
1882.
1883.
1884.
1885.
1886.
1887.
1888.
1889.
1890.
1891.
1892.
1893.
1894.
1895.
1896.
1897.
1898.
1899.
1900.
1901.
1902.
1903.
1904.
1905.
1906.
1907.
1908.
1909.
1910.
1911.
1912.
1913.
1914.
1915.
1916.
1917.
1918.
1919.
1920.
1921.
1922.
1923.
1924.
1925.
1926.
1927.
1928.
1929.
1930.
1931.
1932.
1933.
1934.
1935.
1936.
1937.
1938.
1939.
1940.
1941.
1942.
1943.
1944.
1945.
1946.
1947.
1948.
1949.
1950.
1951.
1952.
1953.
1954.
1955.
1956.
1957.
1958.
1959.
1960.
1961.
1962.
1963.
1964.
1965.
1966.
1967.
1968.
1969.
1970.
1971.
1972.
1973.
1974.
1975.
1976.
1977.
1978.
1979.
1980.
1981.
1982.
1983.
1984.
1985.
1986.
1987.
1988.
1989.
1990.
1991.
1992.
1993.
1994.
1995.
1996.
1997.
1998.
1999.
2000.

```

```

1740. QSTAR(J)=QMAX
1741. CONTINUE
1742. IF(QSTAR(J).GT.0.0)GO TO 12
1743. X(J)=X(J)+DELX(J)
1744. RETURN
1745. CONTINUE
1746. K=0
1747. T=-10.0**34
1748. DO 45 J=1,N
1749. IF(QMAX.LE.QSTAR(J))GO TO 45
1750. TEMP=(C(J)*DELX(J))/(QMAX-QSTAR(J))
1751. IF(TEMP.LE.T)GO TO 45
1752. K=J
1753. T=TEMP
1754. CONTINUE
1755. IF(K.EQ.0)GO TO 10
1756. ***** CHANGE X(K) AND RECOMPUTE INFEASIBILITIES. *****
1757. C
1758. C
1759. C
1760. X(K)=X(K)+DELX(K)
1761. L=J+1
1762. QMAX=QSTAR(K)
1763. IF(DELX(K)/SA.5.16
1764. DO 55 I=1,M
1765. Q(I)=Q(I)-A(I,K)
1766. CONTINUE
1767. GO TO 5
1768. DO 48 I=1,M
1769. Q(I)=Q(I)+A(I,K)
1770. CONTINUE
1771. GO TO 5
1772. C
1773. C
1774. C
1775. C
1776. C
1777. C
1778. C
1779. C
1780. C
1781. C
1782. C
1783. C
1784. C
1785. C
1786. C
1787. C
1788. C
1789. C
1790. C
1791. C
1792. C
1793. C
1794. C
1795. C
1796. C
1797. C
1798. C
1799. C

```

***** NO FURTHER REDUCTION IN INFEASIBILITY CAN BE ATTAINED. *****
 ***** MOVE TO A NEW POINT ON SEARCH SEGMENT. *****
 IF(PHI.EQ.2.0)GO TO 840
 ***** THIS IS PHASE 2. METHOD 1 (METHOD 3 IN REFERENCE *****
 ***** 1 (P. 610). INCREASE ALPHA BY THE MINIMUM *****
 ***** AMOUNT REQUIRED TO OBTAIN A DIFFERENT ROUNDED *****
 ***** SOLUTION THAN AT XL. *****
 DO 830 J=1,N
 TEMP=XTWO(J)-XCNE(J)
 IF(TEMP)831,832,833
 QSTAR(J)=10.0**34
 GO TO 830
 IF(XL(J).LT.0.5)GO TO 832
 QSTAR(J)=-(XL(J)+.5001-AINT(XL(J)+.5))/TEMP
 GO TO 830
 QSTAR(J)=(AINT(XL(J)+.5)+.5001-XL(J))/TEMP
 CONTINUE
 QMAX=10.0**34
 DO 835 J=1,N
 IF(QSTAR(J).LT.QMAX)QMAX=QSTAR(J)
 ALPHA=ALPHA+QMAX
 GO TO 841
 ***** THIS IS PHASE 2. METHOD 2. INCREASE ALPHA BY A *****
 ***** FIXED INCREMENT ZZ. *****


```

1220. IF(C(J)*A(I,J))118,113,117
1221. T=S(I)/ABS(A(I,J))
1222. IF(METHOD.EQ.2)GO TO 942
1223.
1224. ***** THIS IS PHASE 3, PART 2, METHOD 1. THE INTEGER
1225. ***** RESTRICTION IS USED TO COMPUTE HOW MUCH A VARIABLE CAN
1226. ***** CHANGE AND REMAIN FEASIBLE. *****
1227.
1228. IF(T)120,122,121
1229. IF(T.FO.AINT(T))GO TO 122
1230. T=AIN(T)-1.0
1231. GO TO 122
1232. T=AIN(T)
1233. GO TO 122
1234.
1235. ***** THIS IS PHASE 3, PART 2, METHOD 2. THE INTEGER
1236. ***** RESTRICTION IS IGNORED IN COMPUTING HOW MUCH A
1237. ***** VARIABLE CAN CHANGE AND REMAIN FEASIBLE. *****
1238.
1239. IF(T-1.0)141,122,122
1240. T=0.0
1241. IF(T.LT.TEMP)TEMP=T
1242. CONTINUE
1243. IF(TT.GT.A95(C(J))TEMP)GO TO 95
1244. TT=ARS(C(J))TEMP
1245. K=J
1246. CONTINUE
1247. IF(TT)130,127,130
1248. RETURN
1249.
1250. ***** CHANGE X(K) AND RECOMPUTE SLACKS. *****
1251.
1252. CONTINUE
1253. L SOLN2=L SOLN2+1
1254. IF(C(K))131,131,132
1255. X(K)=X(K)-1.0
1256. DO 133 I=1,N
1257. S(I)=S(I)+A(I,K)
1258. GO TO 111
1259. X(K)=X(K)+1.0
1260. DO 137 I=1,M
1261. S(I)=S(I)-A(I,K)
1262. GO TO 111
1263. END
1264.
1265.
1266.
1267.
1268.
1269.
1270.
1271.
1272.
1273.
1274.
1275.
1276.
1277.
1278.
1279.

```

SUBROUTINE PART3

```

***** INITIALIZE THE SECOND MODE OF SEARCH. THIS MODE SEARCHES FOR
***** BETTER SOLUTIONS THAT CAN BE OBTAINED BY CHANGING TWO VARIABLES
***** SIMULTANEOUSLY. FOR EACH PAIR OF VARIABLES X(JPERM(J)) AND
***** X(JPERM(K)) THE ONLY CHANGES CONSIDERED IN THE VARIABLE
***** X(JPERM(J)) WITH LARGER ABSOLUTE VALUE OF OBJECTIVE COEFFICIENT
***** IS TO ADD OR SUBTRACT ONE. SINCE THE ONLY CHANGES THAT NEED BE
***** CONSIDERED IN THE OTHER VARIABLE ARE THOSE THAT STILL EFFECT A

```

```

1930.      * NET INCREASE IN OBJECTIVE FUNCTION VALUE. THE LOWER BOUND ON SJC4 *
1931.      * CHANGES IS STORED IN CAPR(J,K); THE UPPER BOUND IN CAPR(K,J). *
1932.      * .....
1933.      * .....
1934.      * .....
1935.      * .....
1936.      * .....
1937.      * .....
1938.      * .....
1939.      * .....
1940.      * .....
1941.      * .....
1942.      * .....
1943.      * .....
1944.      * .....
1945.      * .....
1946.      * .....
1947.      * .....
1948.      * .....
1949.      * .....
1950.      * .....
1951.      * .....
1952.      * .....
1953.      * .....
1954.      * .....
1955.      * .....
1956.      * .....
1957.      * .....
1958.      * .....
1959.      * .....
1960.      * .....
1961.      * .....
1962.      * .....
1963.      * .....
1964.      * .....
1965.      * .....
1966.      * .....
1967.      * .....
1968.      * .....
1969.      * .....
1970.      * .....
1971.      * .....
1972.      * .....
1973.      * .....
1974.      * .....
1975.      * .....
1976.      * .....
1977.      * .....
1978.      * .....
1979.      * .....
1980.      * .....
1981.      * .....
1982.      * .....
1983.      * .....
1984.      * .....
1985.      * .....
1986.      * .....
1987.      * .....
1988.      * .....
1989.      * .....
1990.      * .....
1991.      * .....
1992.      * .....
1993.      * .....
1994.      * .....
1995.      * .....
1996.      * .....
1997.      * .....
1998.      * .....
1999.      * .....
2000.      * .....
2001.      * .....
2002.      * .....
2003.      * .....
2004.      * .....
2005.      * .....
2006.      * .....
2007.      * .....
2008.      * .....
2009.      * .....
2010.      * .....
2011.      * .....
2012.      * .....
2013.      * .....
2014.      * .....
2015.      * .....
2016.      * .....
2017.      * .....
2018.      * .....
2019.      * .....
2020.      * .....
2021.      * .....
2022.      * .....
2023.      * .....
2024.      * .....
2025.      * .....
2026.      * .....
2027.      * .....
2028.      * .....
2029.      * .....
2030.      * .....
2031.      * .....
2032.      * .....
2033.      * .....
2034.      * .....
2035.      * .....
2036.      * .....
2037.      * .....
2038.      * .....
2039.      * .....

```

```

      DIMENSION A(121,362),B(121),C(121),CAPR(121,121),CPRM(121),
      XD(LX(121),DELTA(121),JPERM(121),LO(121),Q(121),OSTAR(121),
      XS(121),SPRIME(121),X(121),XL(121),XCNE(121),XTWO(121),
      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7
      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

```

      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7

```

```

      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

```

      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7

```

```

      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

```

      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7

```

```

      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

```

      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7

```

```

      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

```

      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7

```

```

      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

```

      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7

```

```

      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

```

      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7

```

```

      L=NNZERO-1
      DO 147 J=1,LX
        LL=J+1
        LJ=JPERM(J)
        DO 141 K=LL,NZERO
          LK=JPERM(K)
          TEMP=ABS(C(LJ)/C(LK))
          IF (TEMP-1.0)142,142,143
          CAPR(J,K)=AINT(TEMP-1.0)
          GO TO 144
          IF (AINT(TEMP)-EQ,TEMP)GO TO 142
          CAPR(J,K)=AINT(TEMP-1.0)+1.0
          IF (TEMP+1.0)145,145,146
          CAPR(K,J)=AINT(TEMP+1.0)
          GO TO 141
          IF (AINT(TEMP)-EQ,TEMP)GO TO 146
          CAPR(K,J)=AINT(TEMP+1.0)-1.0
          CONTINUE
        CONTINUE
      RETURN
      END

```

```

142      GO TO 144
143      CAPR(J,K)=AINT(TEMP-1.0)+1.0
144      IF (TEMP+1.0)145,145,146
145      CAPR(K,J)=AINT(TEMP+1.0)
146      GO TO 141
147      CONTINUE

```

SUBROUTINE PART4

```

* .....
* THE FIRST CHANGE IN X(JPERM(J)) CONSIDERED IS THAT ONE WHICH WOULD
* INCREASE THE OBJECTIVE FUNCTION VALUE. BEFORE CHECKING FOR
* SIMULTANEOUS CHANGES IN X(JPERM(K)) TEST FOR FEASIBILITY.
* .....

```

```

      DIMENSION A(121,362),B(121),C(121),CAPR(121,121),CPRM(121),
      XD(LX(121),DELTA(121),JPERM(121),LO(121),Q(121),OSTAR(121),
      XS(121),SPRIME(121),X(121),XL(121),XCNE(121),XTWO(121),
      COMMON A,ALPHA,B,C,CAPR,CPRM,DELTA,I,IMPROV,DELTA,INVEST,J,
      XHIGH,J,JPERM,K,KEY,LESS,LJ,LK,LL,LM,LML,LQ,LTEMP,M,METHOD,
      XNLESS,NZERO,PHI,C,OMAX,OSTAR,S,SLPRIM,SLSUBK,SPRIME,SUM,T,
      XTEMP,TT,X,XL,XCNE,XTWO,USUBK,ZM,ZZ,N,LSOLN2,LSOLN4,LSOLN5,
      LSOLN6,LSOLN7
      J=J
      LL=JPERM(J)
      CONTINUE

```

```

      J=J
      LL=JPERM(J)
      CONTINUE

```

```

      J=J
      LL=JPERM(J)
      CONTINUE

```


APPENDIX II

SAMPLE INPUT FOR COMPUTER PROGRAM HEUR

APPENDIX III

SAMPLE OUTPUT FOR COMPUTER PROGRAM HEUR

THE CONSTRAINT MATRIX $A(I,J)$ IS:

THE CONTAINING MATERIALS AT (U) IS:														
2.00	40.00	33.00	-36.00	-1.00	43.00	56.00	8.00	6.00	-38.00	12.00	9.00	-16.00	47.00	33.00
5.00	10.00	49.00	17.00	26.00	28.00	38.00	50.00	17.00	35.00	29.00	-37.00	23.00	-33.00	46.00
10.00	10.00	16.00	34.00	-30.00	47.00	13.00	-39.00	-25.00	-17.00	12.00	27.00	50.00	39.00	3.00
5.00	34.00	-31.00	-21.00	34.00	35.00	-15.00	37.00	11.00	-12.00	-15.00	35.00	-32.00	-11.00	-33.00
10.00	4.00	-23.00	15.00	10.00	-40.00	27.00	28.00	-14.00	-21.00	-40.00	-39.00	56.00	-16.00	30.00
5.00	-22.00	-24.00	22.00	44.00	-33.00	-10.00	58.00	-11.00	47.00	55.00	-24.00	5.00	-30.00	28.00
-2.00	8.00	16.00	-12.00	27.00	15.00	16.00	-23.00	26.00	31.00	10.00	22.00	35.00	-30.00	-31.00
5.00	30.00	43.00	5.00	-36.00	-14.00	0.0	0.0	39.00	18.00	38.00	-36.00	59.00	51.00	-33.00
5.00	12.00	-12.00	-25.00	-24.00	-34.00	58.00	-27.00	38.00	-30.00	11.00	29.00	2.00	40.00	31.00
5.00	25.00	-40.00	50.00	-3.00	9.00	5.00	-36.00	21.00	28.00	29.00	-28.00	56.00	16.00	35.00
-2.00	-17.00	0.0	49.00	-35.00	24.00	34.00	33.00	-11.00	34.00	42.00	49.00	56.00	4.00	3.00
5.00	-18.00	13.00	-5.00	36.00	20.00	46.00	36.00	53.00	-16.00	14.00	29.00	31.00	5.00	-9.00
5.00	21.00	39.00	-6.00	35.00	-7.00	41.00	34.00	48.00	8.00	-37.00	-31.00	15.00	48.00	55.00
5.00	-20.00	27.00	27.00	25.00	-28.00	16.00	-8.00	43.00	59.00	-11.00	4.00	14.00	3.00	42.00
5.00	17.00	-32.00	-21.00	47.00	-37.00	-15.00	20.00	-22.00	10.00	5.00	34.00	49.00	-27.00	-29.00
THE RIGHT HAND SIDE ALL IS:														
833.00	978.00	622.00	812.00	752.00	822.00	588.00	911.00	915.00	979.00	584.00	810.00	720.00	590.00	
THE COST COEFFICIENTS C(J) ARE:														
6.00	60.00	63.00	-6.00	71.00	56.00	-14.00	53.00	43.00	-10.00	3.00	10.00	-4.00	52.00	

THE RIGHT HAND SIDE OF (1) IS:

THE COST COEFFICIENTS C(I,J) ARE:

60.00
THE NORMALIZATION FACTOR FOR

THE SIMPLEX METHOD GIVES US:
THE LP SOLUTION HERE IS:

2.91693	1.25460
0.0	0.0

0.0	0.0	0.75636	123.86716
0.0	0.0	0.75636	123.86716

13

10
FOLLOWING. IN ORDER OF INCREASING I, IS THE ITH

THE NORMALIZED INVERSE OF THE OPTIMAL BASIS IS:

	0.0	0.0	6.33003	0.0
	0.0	0.0	0.0	-8.91
	0.0	-0.09148		

-1.60377	0.19445	0.0	0.0
0.0	0.0	-1.40126	0.0

[illegible][illegible][illegible]

15152.2	0.0	-1.663.1	0.90
---------	-----	----------	------

116.79 0.89 0.0 7.29 0.0 -1.27 0.0 0.0 0.0 -2.16 0.0 0.0 0.0 0.0 0.77

AT THE END OF PHASE 2 WE HAVE THE FOLLOWING DATA:
 THE VALUE OF ALPHA WHERE THE FEASIBLE SOLUTION WAS FOUND = 0.0
 THE NUMBER OF POINTS MOVED TO ON THE SEARCH LINE = 0
 THE NUMBER OF TRIAL SOLUTIONS BEFORE FINDING THE FEASIBLE SOLUTION = 4
 THE SOLUTION FROM PHASE 2 IS:

121.00 1.00 0.0 6.00 0.0 0.0 0.0 0.0 0.0 3.0 0.0 0.0 0.0 0.0 0.0

FOR THE FIRST TIME THROUGH MODE 1 WE HAVE THE FOLLOWING DATA:
 THE NUMBER OF SOLUTION CHANGES = 0
 THE NORMALIZED IMPROVEMENT IN THE OBJECTIVE FUNCTION VALUE = 0.0
 THE FOLLOWING OCCURRED PRIOR TO THE TERMINATION OF PHASE 3:
 DURING PHASE 3 WE ENTERED MODE 1 2 TIMES AND HAD 0 SOLUTION CHANGES.
 THE NUMBER OF TIMES THROUGH PART 4 = 32
 THE NUMBER OF SOLUTION CHANGES DURING PART 4 = 0
 THE NUMBER OF TIMES THROUGH PART 5 = 88
 THE NUMBER OF SOLUTION CHANGES DURING PART 5 = 0
 THE NUMBER OF TIMES THROUGH PART 6 = 32
 THE NUMBER OF SOLUTION CHANGES DURING PART 6 = 0
 THE NUMBER OF TIMES THROUGH PART 7 = 30
 THE NUMBER OF SOLUTION CHANGES DURING PART 7 = 0
 AT THE END OF THE HEURISTIC PROCEDURE WE HAVE THE FOLLOWING RESULTS:
 AT THE FINAL APPROXIMATE INTEGER SOLUTION IS:

3.00 1.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

THE FEASIBILITY TEST SLACKS ARE:
 1.66 0.49 0.06 4.05 0.41 0.07 4.46 0.03 4.79 0.07 0.07 0.22 0.48 0.61
 THE OBJECTIVE FUNCTION VALUE OF THE FINAL SOLUTION = 7926.00
 THE NORMALIZED DIFFERENCE BETWEEN THE XENE AND XTO OBJECTIVE FUNCTION VALUES = 4.3132
 THE NORMALIZED IMPROVEMENT BETWEEN THE OPTIMAL LP AND INTEGER OBJECTIVE FUNCTION VALUES = 1.6265
 THE NORMALIZED IMPROVEMENT IN THE OBJECTIVE FUNCTION VALUE SINCE THE FIRST TIME THROUGH MODE 1 = 0.0343
 PHASE 1, METHOD 2, PHASE 2, METHOD 2,
 PHASE 2, CRITERION 4.
 THE CURRENT VALUE OF ALPHA IS 0.0
 THE PHASE 1 SOLUTION XTOX IS:

91.67 0.75 0.0 5.93 0.0 1.79 0.0 0.0 0.0 1.18 0.0 0.0 0.0 0.0 0.77

AT THE END OF PHASE 2 WE HAVE THE FOLLOWING DATA:
 THE VALUE OF ALPHA WHERE THE FEASIBLE SOLUTION WAS FOUND = 0.0
 THE NUMBER OF POINTS MOVED TO ON THE SEARCH LINE = 0
 THE NUMBER OF TRIAL SOLUTIONS BEFORE FINDING THE FEASIBLE SOLUTION = 4
 THE SOLUTION FROM PHASE 2 IS:

131.00 1.00 0.0 6.00 0.0 0.0 0.0 0.0 0.0 3.0 0.0 0.0 0.0 0.0 0.0

FOR THE FIRST TIME THROUGH MODE 1 WE HAVE THE FOLLOWING DATA:
 THE NUMBER OF SOLUTION CHANGES = 0
 THE NORMALIZED IMPROVEMENT IN THE OBJECTIVE FUNCTION VALUE = 0.0
 THE FOLLOWING OCCURRED PRIOR TO THE TERMINATION OF PHASE 3:
 DURING PHASE 3 WE ENTERED MODE 1 2 TIMES AND HAD 0 SOLUTION CHANGES.
 THE NUMBER OF TIMES THROUGH PART 4 = 32
 THE NUMBER OF SOLUTION CHANGES DURING PART 4 = 0
 THE NUMBER OF TIMES THROUGH PART 5 = 88
 THE NUMBER OF SOLUTION CHANGES DURING PART 5 = 0
 THE NUMBER OF TIMES THROUGH PART 6 = 32
 THE NUMBER OF SOLUTION CHANGES DURING PART 6 = 0
 THE NUMBER OF TIMES THROUGH PART 7 = 30
 THE NUMBER OF SOLUTION CHANGES DURING PART 7 = 0
 AT THE END OF THE HEURISTIC PROCEDURE WE HAVE THE FOLLOWING RESULTS:
 AT THE FINAL APPROXIMATE INTEGER SOLUTION IS:

3.00 1.00 0.0 6.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

THE FEASIBILITY TEST SLACKS ARE:
 1.66 0.49 0.06 4.05 0.41 0.07 4.46 0.03 4.79 0.07 0.07 0.22 0.48 0.61
 THE OBJECTIVE FUNCTION VALUE OF THE FINAL SOLUTION = 7926.00
 THE NORMALIZED DIFFERENCE BETWEEN THE XENE AND XTO OBJECTIVE FUNCTION VALUES = 4.3132
 THE NORMALIZED IMPROVEMENT BETWEEN THE OPTIMAL LP AND INTEGER OBJECTIVE FUNCTION VALUES = 1.6265
 THE NORMALIZED IMPROVEMENT IN THE OBJECTIVE FUNCTION VALUE SINCE THE FIRST TIME THROUGH MODE 1 = 0.0343
 PHASE 1, METHOD 2, PHASE 2, METHOD 2,
 PHASE 2, CRITERION 4.
 THE CURRENT VALUE OF ALPHA IS 0.0
 THE PHASE 1 SOLUTION XTOX IS:

91.67 0.75 0.0 5.93 0.0 1.79 0.0 0.0 0.0 1.18 0.0 0.0 0.0 0.0 0.77

AT THE END OF PHASE 2 WE HAVE THE FOLLOWING DATA:
 THE VALUE OF ALPHA WHERE THE FEASIBLE SOLUTION WAS FOUND = 0.0
 THE NUMBER OF POINTS MOVED TO ON THE SEARCH LINE = 0
 THE NUMBER OF TRIAL SOLUTIONS BEFORE FINDING THE FEASIBLE SOLUTION = 4
 THE SOLUTION FROM PHASE 2 IS:

131.00 1.00 0.0 6.00 0.0 0.0 0.0 0.0 0.0 3.0 0.0 0.0 0.0 0.0 0.0

FOR THE FIRST TIME THROUGH MODE 1 WE HAVE THE FOLLOWING DATA:
 THE NUMBER OF SOLUTION CHANGES = 0
 THE NORMALIZED IMPROVEMENT IN THE OBJECTIVE FUNCTION VALUE = 0.0
 THE FOLLOWING OCCURRED PRIOR TO THE TERMINATION OF PHASE 3:
 DURING PHASE 3 WE ENTERED MODE 1 2 TIMES AND HAD 0 SOLUTION CHANGES.
 THE NUMBER OF TIMES THROUGH PART 4 = 32
 THE NUMBER OF SOLUTION CHANGES DURING PART 4 = 0
 THE NUMBER OF TIMES THROUGH PART 5 = 88
 THE NUMBER OF SOLUTION CHANGES DURING PART 5 = 0
 THE NUMBER OF TIMES THROUGH PART 6 = 32
 THE NUMBER OF SOLUTION CHANGES DURING PART 6 = 0
 THE NUMBER OF TIMES THROUGH PART 7 = 30
 THE NUMBER OF SOLUTION CHANGES DURING PART 7 = 0
 AT THE END OF THE HEURISTIC PROCEDURE WE HAVE THE FOLLOWING RESULTS:
 AT THE FINAL APPROXIMATE INTEGER SOLUTION IS:


3.00 1.00 0.0 6.00 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

THE FEASIBILITY TEST SLACKS ARE:
 1.66 0.49 0.06 4.05 0.41 0.07 4.46 0.03 4.79 0.07 0.07 0.22 0.48 0.61
 THE OBJECTIVE FUNCTION VALUE OF THE FINAL SOLUTION = 7926.00
 THE NORMALIZED DIFFERENCE BETWEEN THE XENE AND XTO OBJECTIVE FUNCTION VALUES = 4.3132
 THE NORMALIZED IMPROVEMENT BETWEEN THE OPTIMAL LP AND INTEGER OBJECTIVE FUNCTION VALUES = 1.6265
 THE NORMALIZED IMPROVEMENT IN THE OBJECTIVE FUNCTION VALUE SINCE THE FIRST TIME THROUGH MODE 1 = 0.0343
 PHASE 1, METHOD 2, PHASE 2, METHOD 2,
 PHASE 2, CRITERION 4.
 THE CURRENT VALUE OF ALPHA IS 0.0
 THE PHASE 1 SOLUTION XTOX IS:

91.67 0.75 0.0 5.93 0.0 1.79 0.0 0.0 0.0 1.18 0.0 0.0 0.0 0.0 0.77

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER No. 88	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DOCUMENTATION OF A COMPUTER PROGRAM FOR HILLIER'S HEURISTIC PROCEDURE IN INTEGER LINEAR PROGRAMMING		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL REPORT
7. AUTHOR(s) NANCY E. JACOMIN		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS DEPT. OF OPERATIONS RESEARCH STANFORD UNIVERSITY, STANFORD, CA.		8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0418
11. CONTROLLING OFFICE NAME AND ADDRESS OPERATIONS RESEARCH PROGRAM OFFICE OF NAVAL RESEARCH CODE 434 ARLINGTON, VA. 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-061
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE JULY 1979
		13. NUMBER OF PAGES 55
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		16a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This research was supported in part by National Science Foundation Grant NSF Number - MCS 76-81259 A01, Dept. of Operations Research Stanford University.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) INTEGER PROGRAMMING, HEURISTICS, SOFTWARE		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report gives documentation for the computer code HEUR for finding good approximate solutions to integer linear programming problems using Hillier's heuristic procedure. A listing of the program with sample input and output is included. 		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 6102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)